



VIT[®]

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

Fall Semester 2025-26

Lab Assignment – 2

Slot: L13+L14

Class: VL2025260105679

Branch: B.tech CSBS

Course code & title: CBS3005

Cloud, Microservices and Applications

Faculty name: Nithya K

DA by:

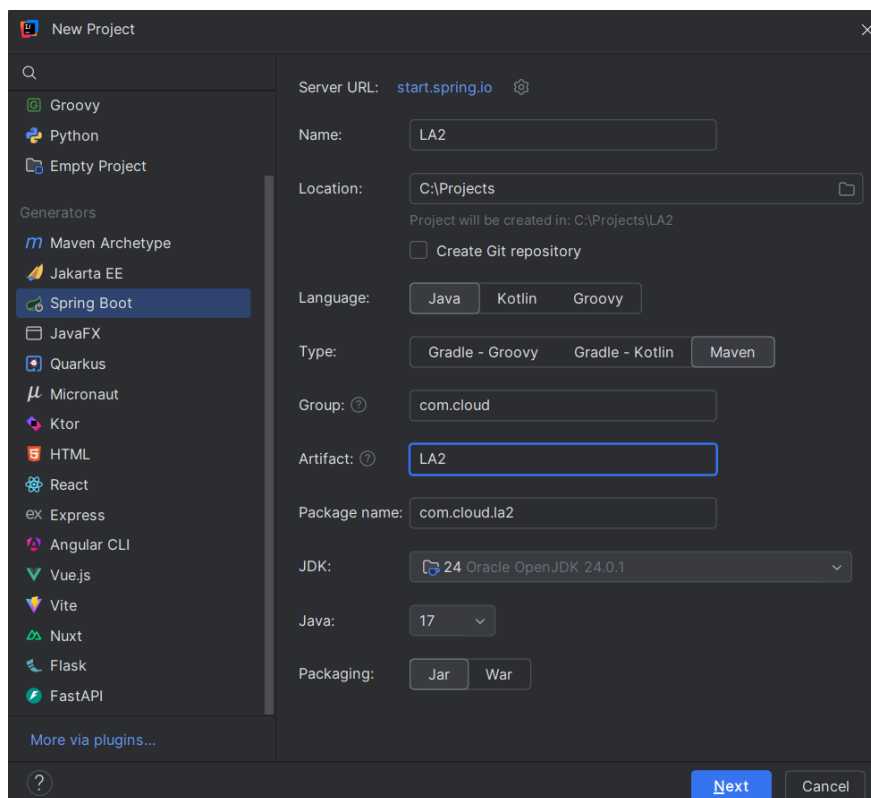
Kartikey Gupta

22BBS0105

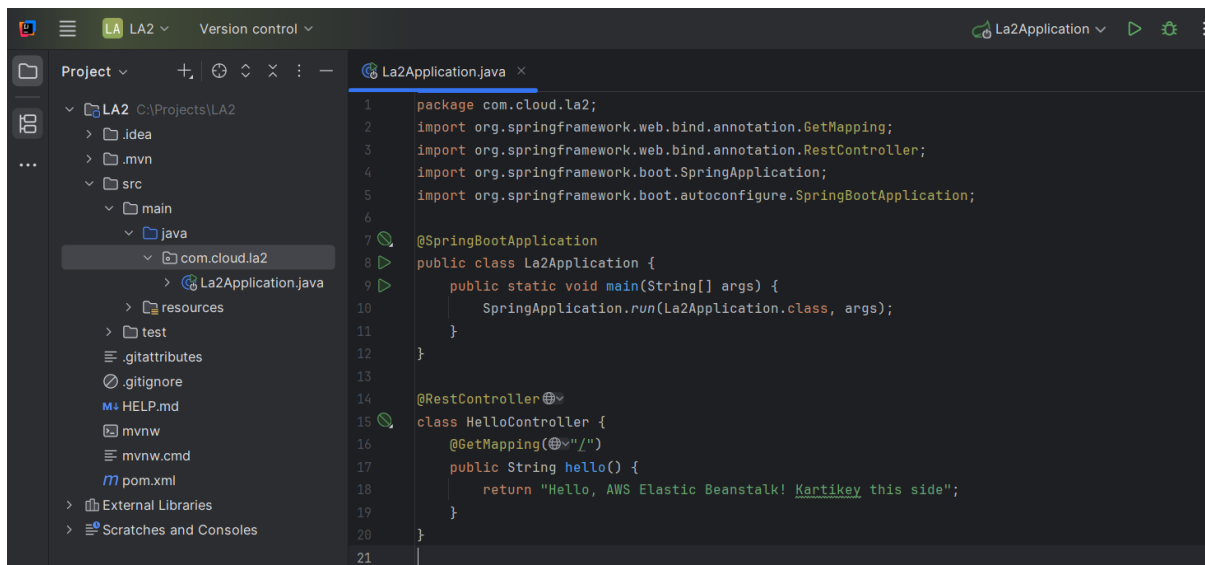
- (i) Create a simple web application using your preferred programming language and framework (e.g., Node.js, Python, Java etc.). Ensure the application is fully functional and ready for deployment. Initialize an AWS Elastic Beanstalk environment for your application. Choose the appropriate platform (e.g., Node.js, Python, Java) and configure the environment settings. Package your web application and deploy it to the Elastic Beanstalk environment. Access the deployed web application via the Elastic Beanstalk URL provided. Test its functionality to confirm that the deployment was successful and that the application is accessible and performs as expected.

Step 1: Create a Simple Spring Boot Application

- i. Open in your IDE IntelliJ, or VS Code.
- ii. Create new springboot project
 - Project: **Maven**
 - Language: **Java**
 - Spring Boot: latest stable version
 - Dependencies: **Spring Web**
- iii. Group: com.cloud, Artifact: LA2

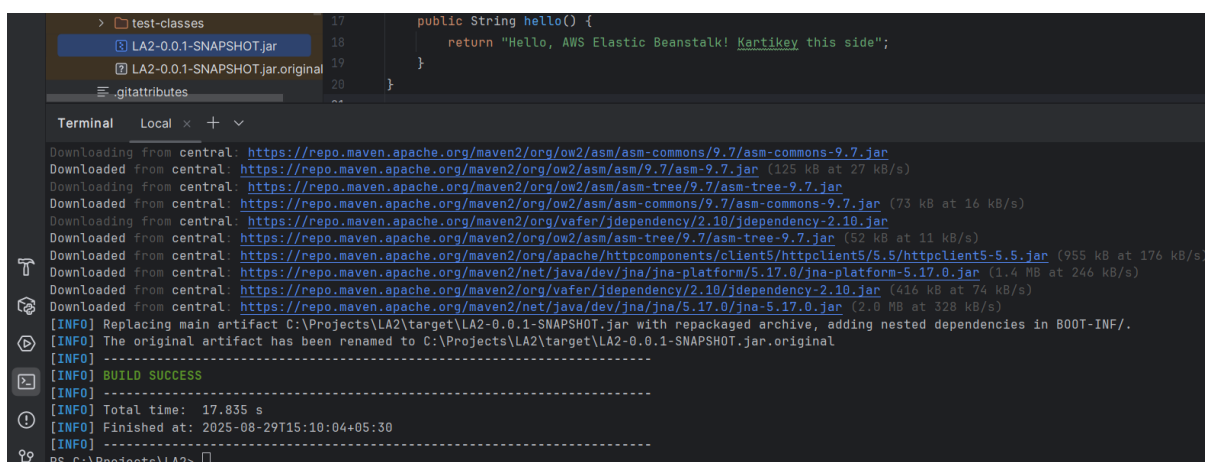


- iv. Edit src/main/java/com/cloud/la2/La2Application.java to create a simple REST endpoint:



```
1 package com.cloud.la2;
2 import org.springframework.web.bind.annotation.GetMapping;
3 import org.springframework.web.bind.annotation.RestController;
4 import org.springframework.boot.SpringApplication;
5 import org.springframework.boot.autoconfigure.SpringBootApplication;
6
7 @SpringBootApplication
8 public class La2Application {
9     public static void main(String[] args) {
10         SpringApplication.run(La2Application.class, args);
11     }
12 }
13
14 @RestController
15 class HelloController {
16     @GetMapping("/")
17     public String hello() {
18         return "Hello, AWS Elastic Beanstalk! Kartikey this side!";
19     }
20 }
21
```

- v. Clean the project using mvn clean package
vi. Build success and Jar file created



```
17 public String hello() {
18     return "Hello, AWS Elastic Beanstalk! Kartikey this side!";
19 }
20 }
21
Terminal Local x + v
Downloading from central: https://repo.maven.apache.org/maven2/org/ow2/asm/asm-commons/9.7/asm-commons-9.7.jar
Downloaded from central: https://repo.maven.apache.org/maven2/org/ow2/asm/asm-commons/9.7/asm-commons-9.7.jar (125 kB at 27 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/ow2/asm/asm-tree/9.7/asm-tree-9.7.jar
Downloaded from central: https://repo.maven.apache.org/maven2/org/ow2/asm/asm-tree/9.7/asm-tree-9.7.jar (73 kB at 16 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/ow2/asm/asm/9.7/asm-9.7.jar
Downloaded from central: https://repo.maven.apache.org/maven2/org/ow2/asm/asm/9.7/asm-9.7.jar (52 kB at 11 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/httpcomponents/client5/httpclient5/5.5/httpclient5-5.5.jar
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/httpcomponents/client5/httpclient5/5.5/httpclient5-5.5.jar (955 kB at 176 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/net/java/dev/jna/jna-platform/5.17.0/jna-platform-5.17.0.jar
Downloaded from central: https://repo.maven.apache.org/maven2/net/java/dev/jna/jna-platform/5.17.0/jna-platform-5.17.0.jar (1.4 MB at 246 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/ow2/asm/asm-tree/9.7/asm-tree-9.7.jar
Downloaded from central: https://repo.maven.apache.org/maven2/org/ow2/asm/asm-tree/9.7/asm-tree-9.7.jar (416 kB at 74 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/net/java/dev/jna/jna/5.17.0/jna-5.17.0.jar
Downloaded from central: https://repo.maven.apache.org/maven2/net/java/dev/jna/jna/5.17.0/jna-5.17.0.jar (2.0 MB at 328 kB/s)
[INFO] Replacing main artifact C:\Projects\LA2\target\LA2-0.0.1-SNAPSHOT.jar with repackaged archive, adding nested dependencies in BOOT-INF/.
[INFO] The original artifact has been renamed to C:\Projects\LA2\target\LA2-0.0.1-SNAPSHOT.jar.original
[INFO] BUILD SUCCESS
[INFO] Total time: 17.835 s
[INFO] Finished at: 2025-08-29T15:10:05:30
[INFO]
```

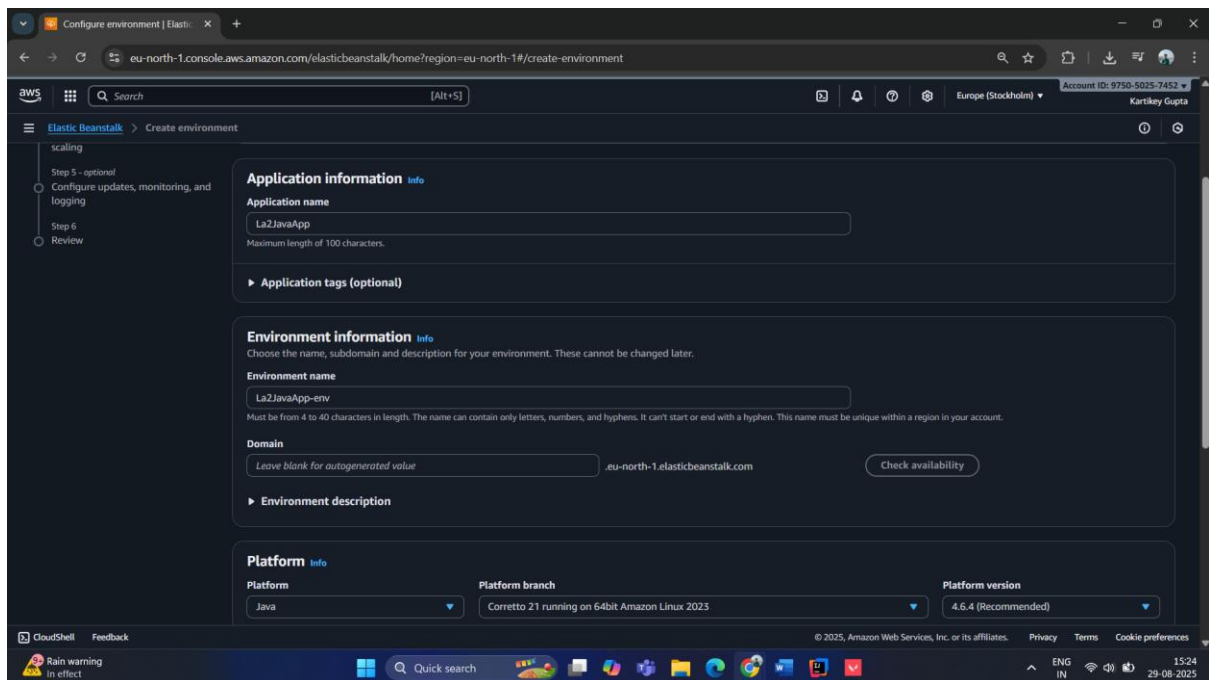
Step 2: Log in to AWS Management Console

- Go to AWS Console.
- Log in with your AWS account.
- Navigate to **Elastic Beanstalk**

Step 3: Create a New Application

- Click **Create application**.
- Enter **Application name**: La2JavaApp.
- Add an optional description: Sample Spring Boot app.

iv. Click **Create application**



Step 5: Configure Environment and Platform

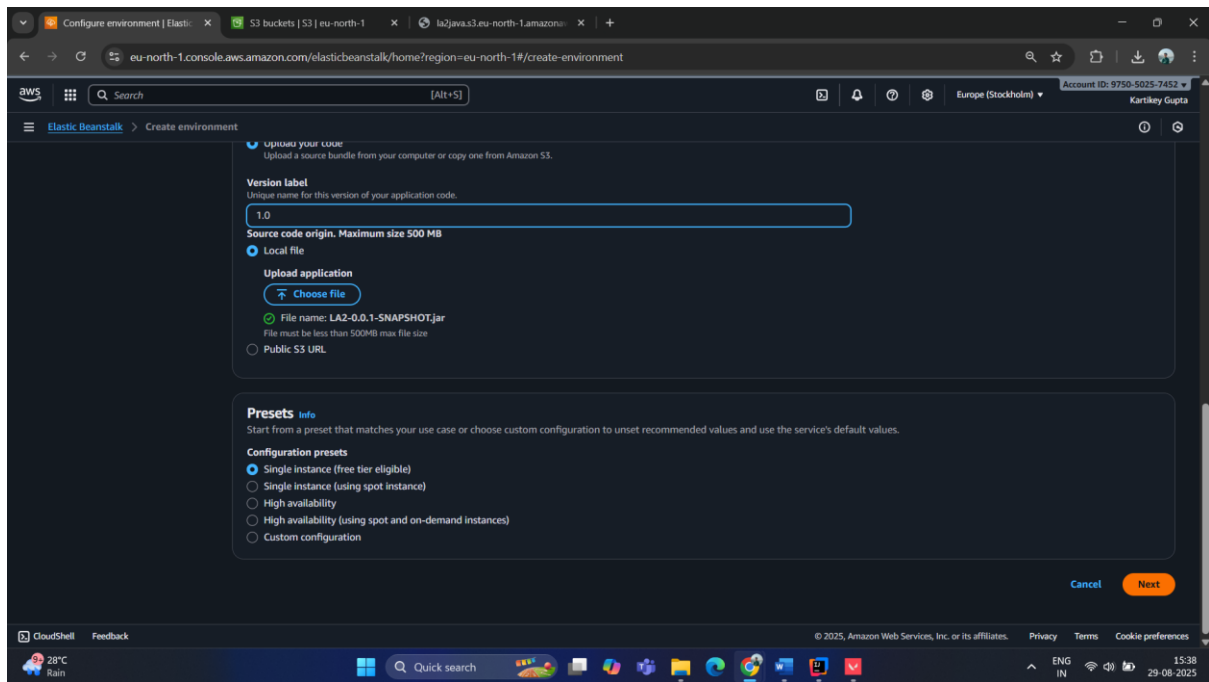
1. **Environment name:** La2JavaApp-env
2. **Platform:** Choose **Java**
 - For Spring Boot JAR: choose **Corretto 17** (or latest Java 17+)
3. **Platform branch:** Keep the default

Step 6: Upload Your Application

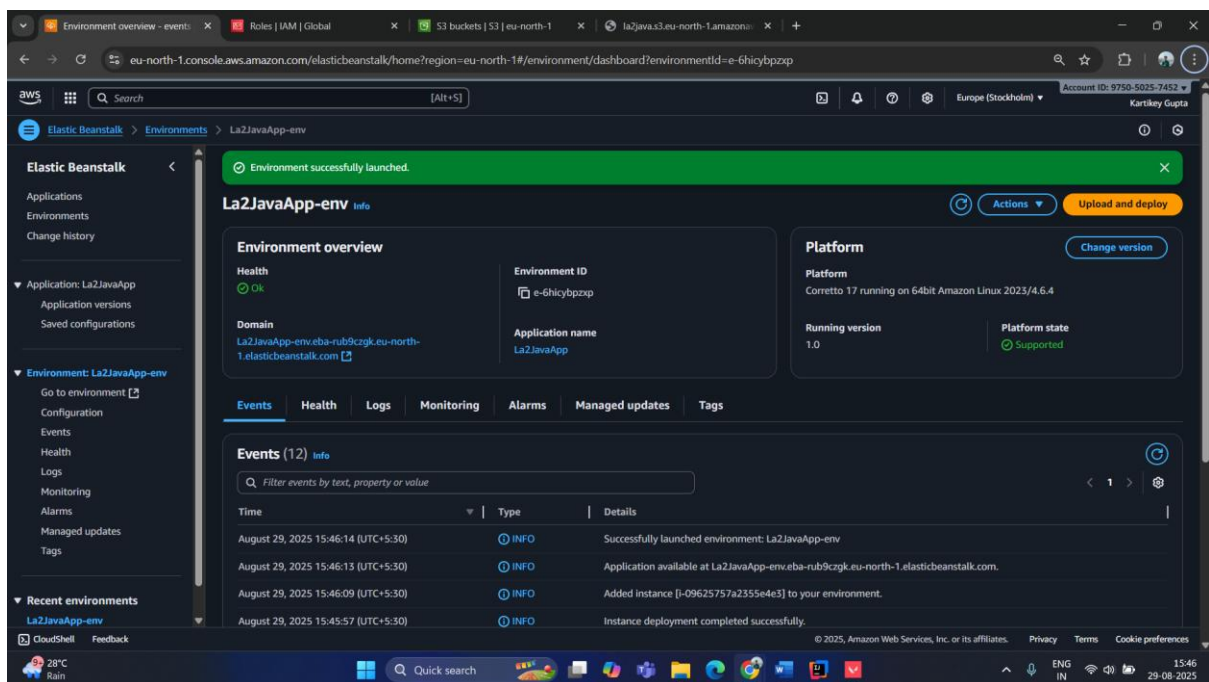
1. Under **Application code**, choose **Upload your code**.
2. Click **Choose file** and select your **JAR file** from target/.

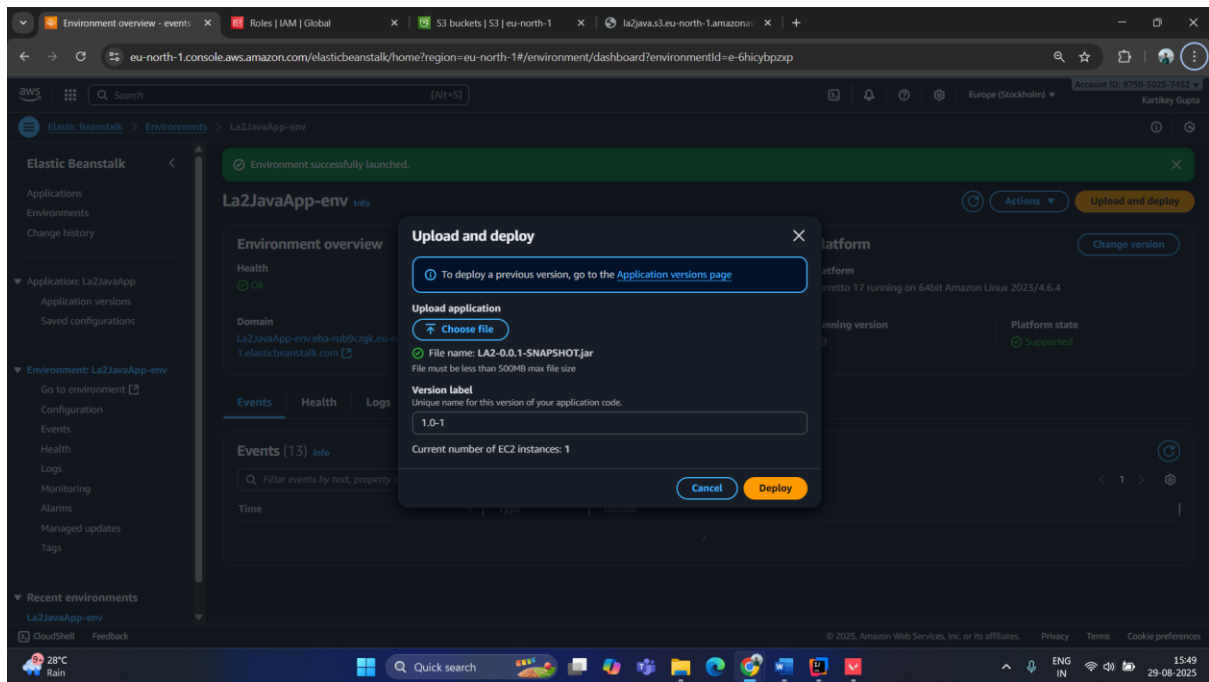
Step 7: Optional Environment Settings

1. **Instance type:** For testing, choose t2.micro (free tier eligible).
2. **Capacity:** Keep single instance.

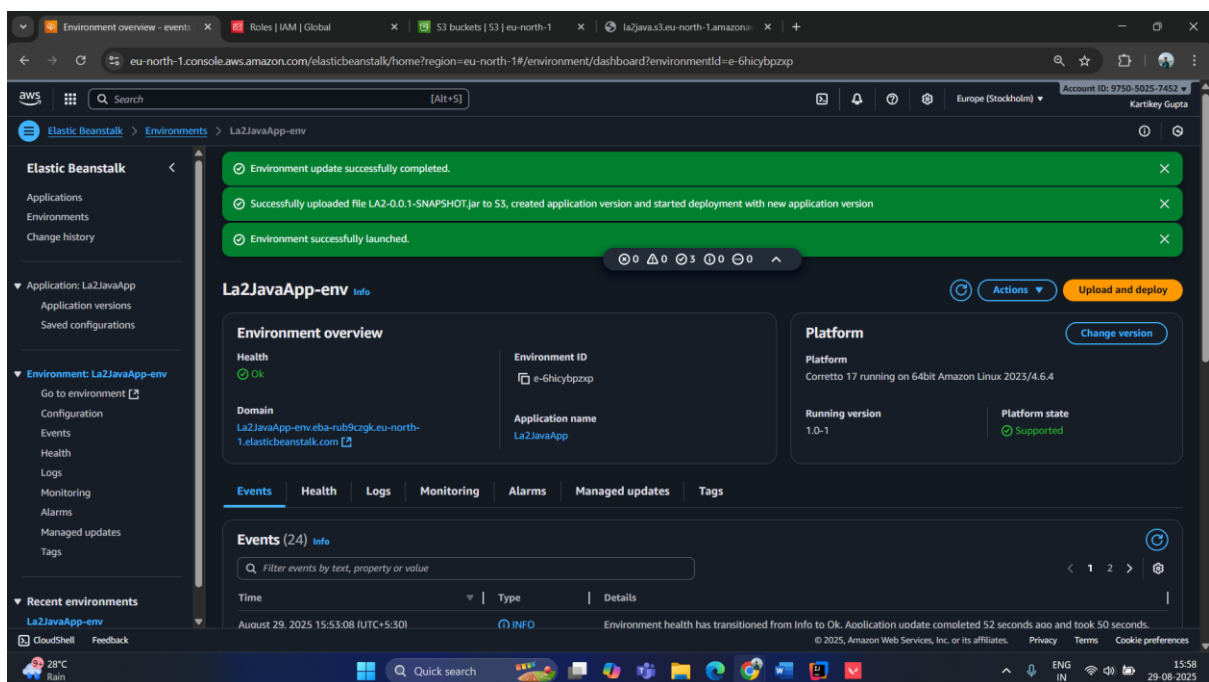


Uploading JAR file:



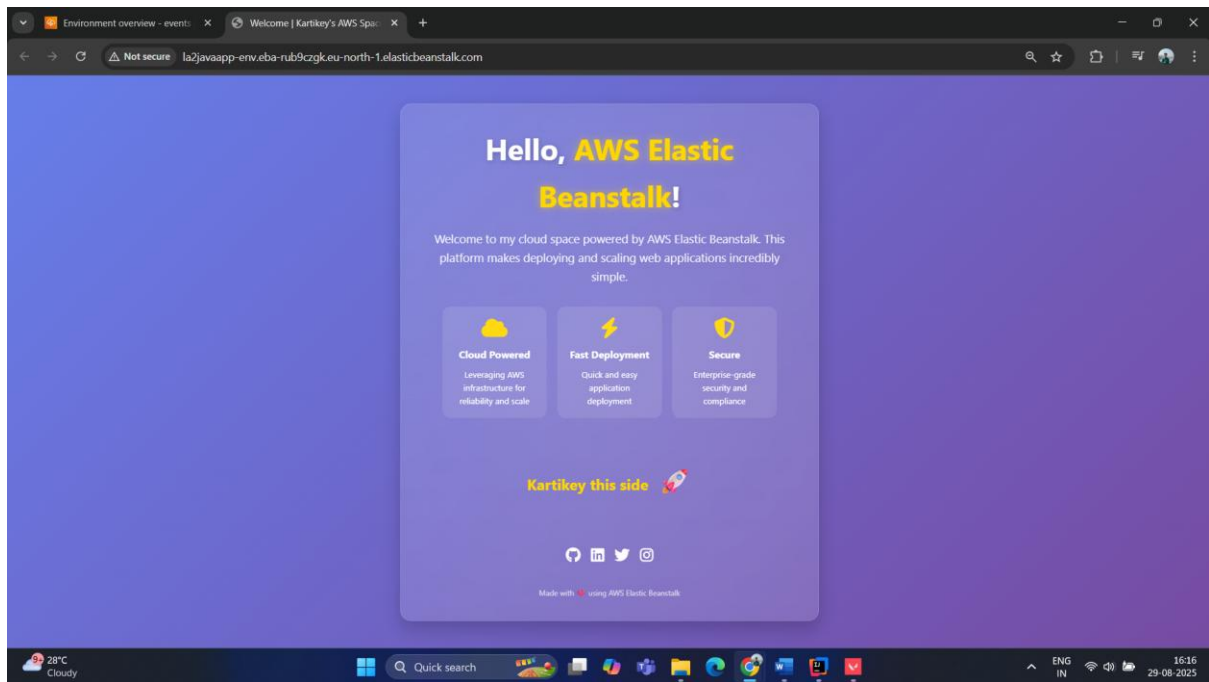


Environment created successfully



Accessing using the link: (will disable later due to incurring charges)

<http://la2javaapp-env.eba-rub9czgk.eu-north-1.elasticbeanstalk.com/>



Deployed web application successfully via the AWS Elastic Beanstalk!!